AMENDMENT(S) TO THE CLAIMS:

The following listing of claims will replace all prior versions, and listings, of claims on the application. Claims being amended are set forth in a larger font than all other claims. All claims are set forth below with one of the following annotations.

- (Original): Claim filed with the application following the specification.
- (Currently amended): Claim being amended in the current amendment paper.
- (Cancelled): Claim cancelled or deleted from the application.
- (Withdrawn): Claim still in the application, but in a non-elected status.
- (New): Claim being added in the current amendment paper.
- (Previously presented): Claim not being currently amended, but which was amended or was new in a previous amendment paper.
- (Not entered): Claim presented in a previous amendment, but not entered or whose entry status unknown. No claim text is shown.

1. (Currently amended) A MAC controller for a wireless station of a wireless network to operate according to one or more transmission policies of a MAC protocol, a transmission according to a policy decomposable into a set of sequential sub-sequences, executing each sub-sequence including transmitting a data unit and in the case that the data unit transmission has a set of one or more expected responses, successfully receiving or not one of the expected responses at the wireless station, each sub-sequence having a finite set of outcomes, each of the outcomes leading to further executing of none or more sub-sequences, such that a transmission from any point in time according to a policy is representable by a tree, each node of the tree representing a sub-sequence, and each branch from a particular node of the tree representing one of the outcomes of the sub-sequence of the particular node, the MAC controller comprising:

a packet scheduler coupled to a data link layer interface and including a scheduling engine and a memory, and

transmit hardware coupled to a physical layer interface, the transmit hardware to execute a sub-sequence and to cause transmission of the data unit of the sub-sequence and to detect which outcome occurs, executing a sub-sequence being describable to the transmit hardware by a corresponding set of execution data, the hardware further including a buffer coupled to the packet scheduler, the buffer organized to receive from the scheduler and store sets of execution data, the hardware further including a signal path to the scheduling engine to provide an indication to the packet scheduler of the outcome of executing a sub-sequence;

wherein the transmit hardware is coupled to a receiver coupled to the physical layer interface to indicate to the transmit hardware the reception at the wireless station of a data unit of one of a set of received data unit types that includes those that may be in response to a transmission of a sub-sequence,

wherein the packet scheduler is arranged to initially prepare and load into the buffer the execution data set for executing all the sub-sequences of a sub-tree starting from the first sub-sequence of the tree;

wherein the buffer is organized as a pipeline for execution data sets; and

wherein after a number of sub-sequences are executed, the packet scheduler, in response to an indication received from the transmit hardware of a particular outcome of a particular sub-sequence, is to provide and load into the buffer the execution data sets needed so that while the hardware is executing the sub-sequence resulting from the particular outcome, the buffer contains the execution data sets of at least each sub-sequence that follows the sub-sequence resulting from the particular outcome,

such that the time available for the packet scheduler to provide and load the execution data sets in response to the indication is at least the time until the transmit hardware completes executing the sub-sequence following the particular outcome,

whereby the sub-sequence being executed and the following sub-sequences that are available to the transmit hardware during the execution form a sub-tree starting with the node sub-sequence being executed.

2. (Original) A MAC controller as recited in claim 1, wherein the packet scheduler includes a processing system including a processor and the memory, the processor to implement the scheduler tasks by executing instructions loadable into the memory,

wherein the memory is loadable with code to instruct the processor to prepare and initially load into the buffer an execution data set for the sub-tree starting with the first node of the tree, and

wherein the memory is loadable with code to instruct the processing system to provide and load into the buffer, in response to the indication of the particular outcome received at the processing system, all the execution data sets for executing all the sub-sequences that together with the sub-sequence following the particular outcome, form the sub-tree starting with the sub-sequence following the particular outcome.

3. (Original) A MAC controller as recited in claim 2, wherein the sub-tree includes levels for each outcome.

4. (Original) A MAC controller as recited in claim 2, wherein the indication of the particular outcome uses an interrupt, the transmit hardware further including

a DMA controller coupled to the processor memory to control direct memory access from the memory; and

a data FIFO coupled to the processor memory to transfer packets of to-be-transmitted data from the processor memory to the transmit hardware under direct memory access,

wherein to set up the DMA controller to transfer the appropriate data from memory into the data FIFO depending on the status of previous transmissions such that only data units that are actually transmitted are transferred to the data FIFO.

5. (Original) A MAC controller as recited in claim 4, wherein the transmit hardware further includes a transmit engine controller ("TxEngine"), wherein the TxEngine is connected to and accepts data from the pipeline, wherein the TxEngine is further connected to and accepts data from the data FIFO, and is further connected to the DMA controller,

wherein the transmit hardware further includes:

a set of processing engines connected to and controlled by the TxEngine to add information to or modify the data unit of a sub-sequence, the set of engines connected to the physical layer interface, and including:

a timestamp engine connected to the TxEngine to include or not timestamp data in the data unit of a sub-sequence;

a frame control engine connected to the TxEngine to include frame control data in the data unit of a sub-sequence;

a CRC engine connected to the TxEngine to include CRC data in the data unit of a sub-sequence;

a counting engine connected to the TxEngine to accept a backoff request from and provide a backoff result to the TxEngine;

a receive filter in the receiver to provide information to the TxEngine controller when one of the set of particular types of packets directed at the wireless station has been received; and

an acknowledgment engine coupled to the receive filter and to the CRC engine to form an acknowledgment data unit and pass said acknowledgment data unit to the CRC engine.

6. (Original) A MAC controller as recited in claim 1, wherein each sub-sequence is describable by a list of primitives that describe the actions needed for executing the sub-sequence.

7. (Original) A MAC controller as recited in claim 1, wherein the MAC protocol is IEEE 802.11 and wherein one of the policies is DCF.

8. (Original) A MAC controller as recited in claim 7, wherein the sub-sequences for transmitting according to DCF include sub-sequences from the set consisting of a request to send sub-sequence("RTS"), a packet transmit sub-sequence ("Frag(dir)") to transmit a data or management packet, a data fragment broadcast sub-sequence ("Frag(broad)") for execution in an access point to broadcast a data or management packet, and a power save poll sub-sequence ("PsPoll") for execution in a non-access point station, and wherein each sub-sequence from the DCF sub-sequence set includes a success consequence being success in receiving an expected response, or implicit success when no response is expected, and some sub-sequences each further includes a failure consequence being failure to receive an expected response.

9. (Original) A MAC controller as recited in claim 8, wherein each of the sub-sequences is describable by a list of primitives that describe the actions needed for executing the sub-sequence.

10. (Currently Amended) A MAC controller as recited in claim 9, wherein the primitives are selected from a set of primitives that includes:

a primitive ("Access") that causes the transmit hardware to attempt immediate access and then undertakes any necessary backoff;

a primitive ("Backoff") that causes the transmit hardware to undertake backoff without an initial access attempt;

a primitive ("Wait_SIFS") that causes the transmit hardware to wait for a short interframe space;

a primitive ("Start") that causes the transmit hardware to instruct the physical layer interface that includes a transmit mode and a receive mode to start receive to transmit mode turn around and the start transmitting;

a primitive ("CRC") that instructs the transmit hardware how to include CRC information in the data unit of the sub-sequence;

a primitive ("Timestamp") that instructs the transmit hardware whether or not and how to include a timestamp in the data unit of the sub-sequence;

a primitive ("FrameControl") that causes the transmit hardware to set up ~~the frame~~ a frame control part in the data unit of the sub-sequence;

a primitive ("Chunk") that causes the transmit hardware to include some or all of a physical layer protocol data unit for transmission in the data unit of the sub-sequence;

a primitive ("Wait_for_ACK") that causes the transmit hardware to wait for an Ack acknowledgment response;

a primitive ("Wait_for_CTS") that causes the transmit hardware to wait for a Clear to Send ("CTS") response;

a primitive ("Wait_for_PsPoll_Ack") that causes the transmit hardware to wait for acknowledgment to a PsPoll;

a primitive ("Finished") that causes the transmit hardware to initiate a post transmission backoff; and

a primitive ("Cancel") that stops all sub-sequences,

wherein any sub-sequence of the DCF sub-sequences includes:

a primitive from the group consisting of Access, Backoff, and Wait_SIFS,

a Start,

a CRC,

a Timestamp,

a FrameControl, and

a Chunk,

and wherein any sub-sequence of the DCF sub-sequences may include none, one or both of:

a primitive from the group consisting of Wait_for_ACK, Wait_for_CTS, and Wait_for_PsPoll_Ack, and

a Finished.

11. (Original) A MAC controller as recited in claim 7, wherein one of the policies is PCF.

12. (Original) A MAC controller as recited in claim 8, wherein one of the policies is PCF and wherein the sub-sequences for transmitting according to PCF include sub-sequences from the set consisting of:

a contention-free poll sub-sequence ("CfPoll") that may include a piggybacked contention free acknowledgment ("CfAck"), the CfPoll sub-sequence being for execution in an access point;

a contention-free poll with data sub-sequence ("CfPollData"), that may include a piggybacked contention free acknowledgment ("CfAck"), the CfPollData being a CfPoll that further includes a data fragment and being for execution in an access point;

a data packet transmit sub-sequence ("Data(dir)") that may include a piggybacked CfAck;

a management packet sub-sequence ("Mgt(dir)") for execution in an access point;

a data packet broadcast sub-sequence ("Data(broad)") for execution in an access point; and

a management packet broadcast sub-sequence ("Mgt(broad)") for execution in an access point,

and wherein each sub-sequence from the PCF sub-sequence set includes a first success outcome being either success in receiving one of a first set of expected responses or implicit success when no response is expected, wherein some sequences for execution in an access point further include a second success outcome being success in receiving one of a first set of expected responses from a non-access point station, and further receiving an indication that the non-access point station has more data, and wherein some sub-sequences each further include a failure consequence being failure to receive an expected response.

13. (Original) A MAC controller as recited in claim 12, wherein each of the sub-sequences is describable by a list of primitives that describe the actions needed for executing the sub-sequence.

14. (Currently amended) A MAC controller as recited in claim 13,

wherein the set of primitives for each of the DCF sub-sequences includes:

a primitive ("Access") that causes the transmit hardware to attempt immediate access and then undertakes any necessary backoff;

a primitive ("Backoff") that causes the transmit hardware to undertake backoff without an initial access attempt;

a primitive ("Wait_SIFS") that causes the transmit hardware to wait for a short interframe space;

a primitive ("Start") that causes the transmit hardware to instruct the physical layer interface that includes a transmit mode and a receive mode to start receive to transmit mode turn around and the start transmitting;

a primitive ("CRC") that instructs the transmit hardware how to include CRC information in the data unit of the sub-sequence;

a primitive ("Timestamp") that instructs the transmit hardware whether or not and how to include a timestamp in the data unit of the sub-sequence;

a primitive ("FrameControl") that causes the transmit hardware to set up ~~frame~~ a frame control part in the data unit of the sub-sequence;

a primitive ("Chunk") that causes the transmit hardware to include some or all of a physical layer protocol data unit for transmission in the data unit of the sub-sequence;

a primitive ("Wait_for_ACK") that causes the transmit hardware to wait for an acknowledgment ("Ack") response;

a primitive ("Wait_for_CTS") that causes the transmit hardware to wait for a Clear to Send ("CTS") response;

a primitive ("Wait_for_PsPoll_Ack") that causes the transmit hardware to wait for acknowledgment to a PsPoll;

a primitive ("Finished") that causes the transmit hardware to initiate a post transmission backoff; and

a primitive ("Cancel") that stops all sub-sequences,

wherein any sub-sequence of the DCF sub-sequence set includes:

a primitive from the group consisting of Access, Backoff, and Wait_SIFS,

a Start,

a CRC,

a Timestamp,

a FrameControl, and

a Chunk,

wherein any DCF sub-sequence further includes none, one or both of

a primitive from the group consisting of Wait_for_ACK, Wait_for_CTS, and Wait_for_PsPoll_Ack, and

a Finished,

wherein the set of primitives for each of the PCF sub-sequences includes none or more of the primitives for the DCF sub-sequence set,

wherein the Wait_SIFS primitive optionally causes the transmit hardware to piggyback a CfAck,

wherein the Backoff primitive optionally causes the transmit hardware to send a CfAck,

wherein the set of primitives for the PCF sub-sequence set further includes:

a primitive ("Wait_PIFS") that causes the transmit hardware to wait for a point coordination function interframe space;

a primitive ("Wait_for_CfPoll") that causes the transmit hardware of a non-access point station to wait for a CfPoll from an access point addressed to said non-access point station;

a primitive ("Wait_for_CfPoll_Ack") that causes the transmit hardware of an access point to wait for an acknowledgment after transmitting a CfPoll;

a primitive ("Wait_for_CfPollData_Ack") that causes the transmit hardware of an access point to wait for an acknowledgment after transmitting a CfPoll with data;

a primitive ("Wait_for_Data_Ack") that causes the transmit hardware of an access point to wait for an acknowledgment from a non-access point station after transmitting a Data(dir); and

a primitive ("Wait_Beacon_CFP_Access") that causes the transmit hardware to wait for a beacon access opportunity at the start of a contention-free period,

wherein any sub-sequence of the PCF sub-sequence set includes:

a primitive from the group consisting of Access, Wait_SIFS, Wait_PIFS, Backoff, Wait_Beacon_CFP_Access, and Wait_CfPoll,

a Start,

a CRC,

a Timestamp,

a FrameControl, and

a Chunk,

and wherein any PCF sub-sequence further includes none, one or both of

a primitive from the group consisting of Wait_for_PsPoll_Ack, Wait_for_CfPoll_Ack, Wait_for_CfPollData_Ack, Wait_for_ACK, Wait_for_CfAck, and Wait_for_CTS, and

a Finished.

15. (Original) A MAC controller as recited in claim 14, wherein the packet scheduler includes a processing system including a processor and the memory, the processor to implement the scheduler tasks by executing instructions loadable into the memory,

wherein the indication of the particular outcome uses an interrupt, the transmit hardware further including

a DMA controller coupled to the processor memory to control direct memory access therefrom ; and

a data FIFO coupled to the processor memory to transfer packets of to-be-transmitted data from the processor memory to the transmit hardware under direct memory access, and

wherein the transmit hardware is arranged to set up the DMA controller to transfer the appropriate data from memory into the data FIFO depending on the status of previous

transmissions such that only data units that are actually transmitted are transferred to the data FIFO.

16. (Original) A MAC controller as recited in claim 15, wherein the transmit hardware further includes:

a transmit engine controller ("TxEngine") connected to and accepting data from the pipeline, the TxEngine further connected to and accepting data from the data FIFO, and further connected to the DMA controller;

a set of processing engines connected to and controlled by the TxEngine to add information to or modify the data unit of a sub-sequence, the set of engines connected to the physical layer interface, and including:

a timestamp engine connected to the TxEngine controller to include or not timestamp data in the data unit of a sub-sequence;

a frame control engine connected to the TxEngine controller to include frame control data in the data unit of a sub-sequence;

a CRC engine connected to the TxEngine controller to include CRC data in the data unit of a sub-sequence;

a counting engine connected to the TxEngine controller to accept a backoff request from and provide a backoff result to the TxEngine controller;

a receive filter in the receiver to provide information to the TxEngine controller when one of a set of particular types of packets directed at the wireless station has been received; and

an acknowledgment engine coupled to the receive filter and to the CRC engine to form an acknowledgment data unit and pass said acknowledgment data unit to the CRC engine.

17. (Original) A MAC controller for a wireless station of a wireless network to operate according to one or more transmission policies of a MAC protocol, a transmission according to a policy decomposable into a set of sequential sub-sequences, executing each sub-sequence including transmitting a data unit and in the case that the data unit transmission has a set of one or more expected responses, successfully receiving or not one of the expected responses at the wireless station, each sub-sequence having a finite set of outcomes, each of the outcomes leading to further executing of none or more sub-sequences, the MAC controller comprising:

transmit hardware coupled to a physical layer interface, the transmit hardware capable of executing a sub-sequence by causing transmission of the data unit of the sub-sequence and detecting which outcome occurs,

a packet scheduler coupled to a data link layer interface and configured to prepare a set of execution data that describes to the transmit hardware how to execute a sub-sequence, the packet scheduler including a memory,

wherein the transmit hardware includes a buffer coupled to the packet scheduler to receive a set of execution data, and a signal path to the packet scheduler to provide an indication to the packet scheduler of the outcome of executing a sub-sequence,

wherein the transmit hardware is coupled to a receiver coupled to the physical layer interface to indicate to the transmit hardware the reception at the wireless station of a data unit of one of a set of received data unit types that includes those that may be in response to a transmission of a sub-sequence,

wherein executing a sub-sequence is describable to the transmit hardware by a corresponding set of execution data,

wherein the packet scheduler is configured to prepare and pass to the transmit hardware via the buffer the execution data sets for executing one or more sub-sequences in response to an indication from the transmit hardware of the outcome of a sub-sequence execution.

18. (Original) A MAC controller as recited in claim 17, wherein the packet scheduler includes a processing system including a processor and the memory, the processor to implement the scheduler tasks by executing instructions loadable into the memory,

wherein the memory is loadable with code to instruct the processor to prepare and pass to the transmit hardware via the buffer the execution data sets of the one or more sub-sequences.

19. (Original) A MAC controller as recited in claim 17, wherein the buffer is organized as a pipeline for execution data sets, wherein a transmission from any point in time according to a policy is representable by a tree, each node of the tree representing a sub-sequence, and each branch from a particular node of the tree representing one of the outcomes of the sub-sequence of the particular node, and wherein after a number of levels of the tree have been executed, the one or more data sets prepared and passed in response to the indication are such that when the transmit hardware is executing a particular sub-sequence, the transmit hardware already has available the execution data sets of at least one sub-sequence that follows the particular sub-sequence,

whereby the particular sub-sequence being executed and the following sub-sequences that are available to the transmit hardware during the execution form a sub-tree starting with the node of the particular sub-sequence.

20. (Original) A MAC controller as recited in claim 19, wherein the sub-tree has two levels and the execution data set of every outcome sub-sequence of the particular sub-sequence is available to the hardware during the execution of the particular sub-sequence.

21. (Original) A MAC controller as recited in claim 19, wherein the sub-tree has at least two levels for each outcome of the particular sub-sequence being executed.

22. (Original) A MAC controller as recited in claim 19, wherein the sub-tree is a pruned sub-tree that has more levels for the most likely path of sub-sequences of the tree.

23. (Original) In a wireless station of a wireless data network, a method of transmitting packets of information according to a MAC transmission policy, the transmission of the packets according to the policy decomposable into a set of sequential sub-sequences, executing each sub-sequence including transmitting a data unit and in the case that the data unit transmission has a set of one or more expected responses, successfully receiving or not one of the expected responses at the wireless station, each sub-sequence having a finite set of outcomes, each of the outcomes leading to further executing of none or more sub-sequences, the method comprising

(a) determining the next sub-sequence,

(b) passing to a buffer of a transmit engine included in the wireless station a set of execution data for the next sub-sequence describing how to execute the next sub-sequence;

(c) determining for each outcome of the next sub-sequence, at least one of the sub-sequences following the outcome;

(d) passing to the buffer the execution data sets of at least the sub-sequences following each outcome of the next sub-sequence;

(e) waiting for an indication from the transmit engine of the outcome of executing the next sub-sequence, the outcome defining a new next sub-sequence as the corresponding determined sub-sequence following the outcome if the outcome has a following sub-sequence;

(f) in the case that the outcome has a following sub-sequence, continuing by repeating the steps starting with said step (c) of determining for each outcome of the new next sub-sequence, the sub-sequence following the outcome, and in the case that the outcome does not have a following sub-sequence, continuing by repeating the steps starting with said step (a) of determining for a new next sub-sequence,

the method further comprising repeating in the transmit engine until no more execution data is available the steps of

accepting the execution data set of the next sub-sequence from the buffer;

executing the next sub-sequence; and

after one of the outcomes of the next sub-sequence has occurred, accepting the execution data set of the sub-sequence of the outcome from the buffer and providing the indication of the outcome from the transmit engine.

24. (Original) A method as recited in claim 23, wherein the step of determining for each outcome of the next sub-sequence, the sub-sequence following the outcome, and wherein the step of passing to the buffer passes the execution data sets of the sub-sequences following each outcome of the next sub-sequence.

25. (Original) A method as recited in claim 23, wherein each sub-sequence is describable by a list of primitives that describe the actions needed for executing the sub-sequence.

26. (Original) A method as recited in claim 23, wherein the MAC transmission policy is the DCF of the IEEE 802.11 protocol.

27. (Original) A method as recited in claim 26, wherein the MAC transmission policy is the PCF of the IEEE 802.11 protocol.

28. (Original) A method as recited in claim 26, wherein the sub-sequences for transmitting according to DCF include sub-sequences from the set consisting of a request to send sub-sequence("RTS"), a packet transmit sub-sequence ("Frag(dir)") to transmit a data or management packet, a data fragment broadcast sub-sequence ("Frag(broad)") for execution in an access point to broadcast a data or management packet, and a power save poll sub-sequence ("PsPoll") for execution in a non-access point station, and wherein each sub-sequence from the DCF sub-sequence set includes a success consequence being success in receiving an expected response, or implicit

success when no response is expected, and some sub-sequences each further includes a failure consequence being failure to receive an expected response.

29. (Original) A method as recited in claim 28, wherein each of the sub-sequences is describable by a list of primitives that describe the actions needed for executing the sub-sequence.

30. (Original) In a wireless station of a wireless data network, a method of transmitting packets of information according to a MAC transmission policy, the transmission of the packets according to the policy decomposable into a set of sequential sub-sequences, executing each sub-sequence including transmitting a data unit and in the case that the data unit transmission has a set of one or more expected responses, successfully receiving or not one of the expected responses at the wireless station, each sub-sequence having a finite set of outcomes, each of the outcomes leading to further executing of none or more sub-sequences, such that following the policy starting from a first sub-sequence is representable by a tree, each node of the tree representing a sub-sequence, and each branch from a particular node of the tree representing one of the outcomes of the sub-sequence of the particular node, the method comprising:

initially preparing the execution data of the initial sub-sequence and of each sub-sequence of the next level of the tree of the policy;

passing the sets of execution data for the initial and next tree level sub-sequences to a transmit engine for execution;

repeating until the tree has been traversed:

waiting for an indication from the transmit engine of the outcome of completion of execution of a sub-sequence; and

upon receiving the indication, preparing and passing to the hardware the execution data sets of each sub-sequence of the level of the tree that follows the sub-sequence of the outcome,

such that a pipeline of execution data for the transmit engine is formed containing two or more levels of the tree.

31. (Original) A MAC controller for a wireless station of a wireless network to operate according to one or more transmission policies of a MAC protocol, a transmission according to a policy decomposable into a set of sequential sub-sequences, executing each sub-sequence including transmitting a data unit and in the case that the data unit transmission has a set of one or more expected responses, successfully receiving or not one of the expected responses at the wireless station, each sub-sequence having a finite set of outcomes, each of the outcomes leading to further executing of none or more sub-sequences, such that a transmission from any point in time according to a policy is representable by a tree, each node of the tree representing a sub-sequence, and each branch from a particular node of the tree representing one of the outcomes of the sub-sequence of the particular node, the MAC controller comprising:

a packet scheduler coupled to a data link layer interface and including a scheduling engine and a memory, and

transmit hardware coupled to a physical layer interface, the transmit hardware to execute a sub-sequence and to cause transmission of the data unit of the sub-sequence and to detect which outcome occurs, executing a sub-sequence being describable to the transmit hardware by a corresponding set of execution data, the hardware further including a buffer coupled to the packet scheduler, the buffer organized to receive from the scheduler and store a set of execution data, the hardware further including a signal path to the scheduling engine to provide an indication to the packet scheduler of the outcome of executing a sub-sequence;

wherein the transmit hardware is coupled to a receiver coupled to the physical layer interface to indicate to the transmit hardware the reception at the wireless station of a data unit of one of a set of received data unit types that includes those that may be in response to a transmission of a sub-sequence,

wherein the packet scheduler is arranged to initially prepare and load into the buffer the execution data set for executing the first sub-sequence of the tree;

wherein, the packet scheduler, in response to an indication received from the transmit hardware of a particular outcome of a particular sub-sequence, is to provide and load into the buffer the execution data needed for executing one or more sub-sequences that are further along the tree than the particular sub-sequence.

32. (Original) A MAC controller for a wireless station of a wireless network to operate according to one or more transmission policies of a MAC protocol, a transmission according to a policy decomposable into a set of sequential sub-sequences, executing each sub-sequence including transmitting a data unit and in the case that the data unit transmission has a set of one or more expected responses, successfully receiving or not one of the expected responses at the wireless station, each sub-sequence having a finite set of outcomes, each of the outcomes leading to further executing of none or more sub-sequences, such that a transmission from any point in time according to a policy is representable by a tree, each node of the tree representing a sub-sequence, and each branch from a particular node of the tree representing one of the outcomes of the sub-sequence of the particular node, the MAC controller comprising:

means for scheduling packets,

sub-sequence execution means coupled to a physical layer interface, the sub-sequence execution means for executing a sub-sequence by transmitting the data unit of the sub-sequence and detecting which outcome occurs, executing a sub-sequence being describable by a corresponding set of execution data; and

buffer means coupled to the packet scheduler for one or more sets of execution data from the scheduling means,

wherein the execution means is coupled to a receiver coupled to the physical layer interface to indicate to the execution means the reception at the wireless station of a

data unit of one of a set of received data unit types that includes those that may be in response to a transmission of a sub-sequence,

wherein the scheduling means initially prepares and loads into the buffer means the execution data set for executing all the sub-sequences of a sub-tree starting from the first sub-sequence of the tree;

wherein the execution means provides an indication to the scheduling means of the outcome of executing a sub-sequence,

wherein after a number of sub-sequences are executed, the scheduling means, in response to an indication received from the execution means of a particular outcome of a particular sub-sequence, provides and loads into the buffer means one or more execution data sets such that while the hardware is executing the sub-sequence resulting from the particular outcome, the buffer contains the execution data sets of at least each sub-sequence that follows the sub-sequence resulting from the particular outcome,

such that the time available for the scheduling means to provide and load the execution data sets in response to the indication is at least the time until the execution means completes executing the sub-sequence following the particular outcome,

whereby the sub-sequence being executed and the following sub-sequences that are available to the execution means during the execution form a sub-tree starting with the node sub-sequence being executed.

33. (Original) A MAC controller as recited in claim 32, wherein the sub-tree has two levels.

34. (Original) A MAC controller as recited in claim 32, wherein each sub-sequence is describable by a list of primitives that describe the actions needed for executing the sub-sequence.

35. (Original) A MAC controller as recited in claim 32, wherein the MAC transmission policy is the DCF of the IEEE 802.11 protocol.

36. (Original) A MAC controller as recited in claim 35, wherein the MAC transmission policy is the PCF of the IEEE 802.11 protocol.

37. (Original) A MAC controller as recited in claim 35, wherein the sub-sequences for transmitting according to DCF include sub-sequences from the set consisting of a request to send sub-sequence("RTS"), a packet transmit sub-sequence ("Frag(dir)") to transmit a data or management packet, a data fragment broadcast sub-sequence ("Frag(broad)") for execution in an access point to broadcast a data or management packet, and a power save poll sub-sequence ("PsPoll") for execution in a non-access point station, and wherein each sub-sequence from the DCF sub-sequence set includes a success consequence being success in receiving an expected response, or implicit success when no response is expected, and some sub-sequences each further includes a failure consequence being failure to receive an expected response.

38. (Original) A MAC controller as recited in claim 37, wherein each of the sub-sequences is describable by a list of primitives that describe the actions needed for executing the sub-sequence.

39. (Original) A carrier medium carrying computer readable code segments to instruct a processor of a processing system of a wireless station of a wireless data network to carry a packet scheduling method for use in a method for transmitting packets of information according to a MAC transmission policy, the transmission of the packets according to the policy decomposable into a set of sequential sub-sequences, executing each sub-sequence including transmitting a data unit and in the case that the data unit transmission has a set of one or more expected responses, successfully receiving or not one of the expected responses at the wireless station, each sub-sequence having a finite set of outcomes, each of the outcomes leading to further executing of none or more sub-sequences, the carrier medium comprising

(a) one or more code segments to instruct the processor to determine the next sub-sequence,

(b) one or more code segments to instruct the processor to pass to a buffer of a transmit engine included in the wireless station a set of execution data for the next sub-sequence describing how to execute the next sub-sequence;

(c) one or more code segments to instruct the processor to determine for each outcome of the next sub-sequence, at least one of the sub-sequence following the outcome;

(d) one or more code segments to instruct the processor to pass to the buffer the execution data sets of at least the sub-sequences following each outcome of the next sub-sequence;

(e) one or more code segments to instruct the processor to wait for an indication from the transmit engine of the outcome of executing the next sub-sequence, the outcome defining a new next sub-sequence as the corresponding determined sub-sequence following the outcome if the outcome has a following sub-sequence;

(f) in the case that the outcome has a following sub-sequence, one or more code segments to instruct the processor to continue by repeating the steps starting with said step (c) of determining for each outcome of the new next sub-sequence, the sub-sequence following the outcome, and in the case that the outcome does not have a following sub-sequence, one or more code segments to instruct the processor to continue by repeating the steps starting with said step (a) of determining for a new next sub-sequence,

the method for transmitting packets further comprising repeating in the transmit engine until no more execution data is available the steps of

accepting the execution data set of the next sub-sequence from the buffer;

executing the next sub-sequence; and

after one of the outcomes of the next sub-sequence has occurred, accepting the execution data set of the sub-sequence of the outcome from the buffer and providing the indication of the outcome from the transmit engine.